

# La machine d'états de JKI



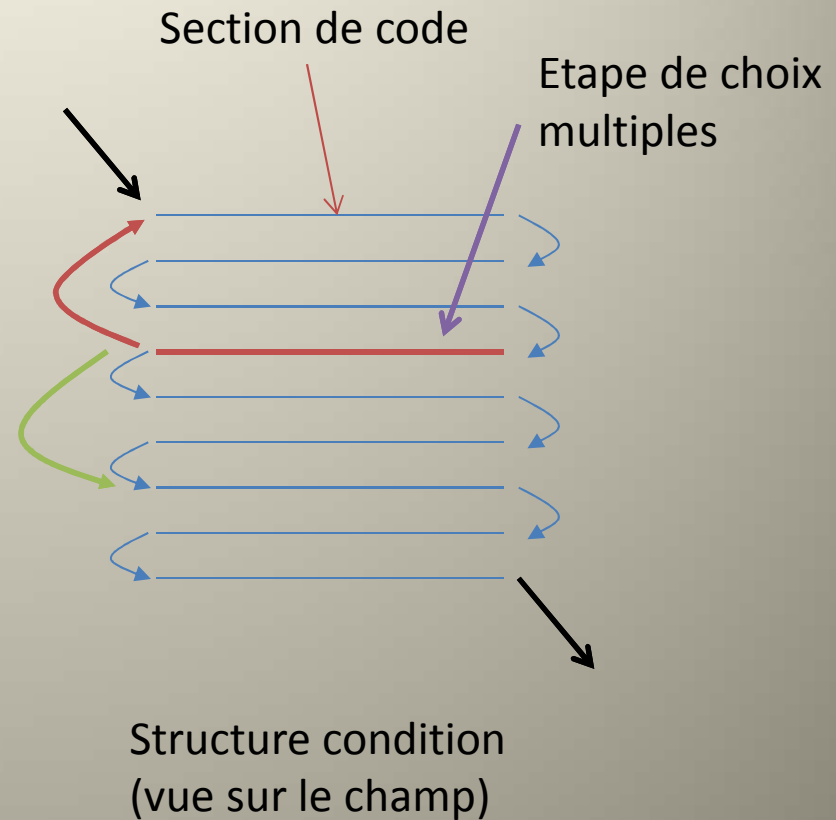
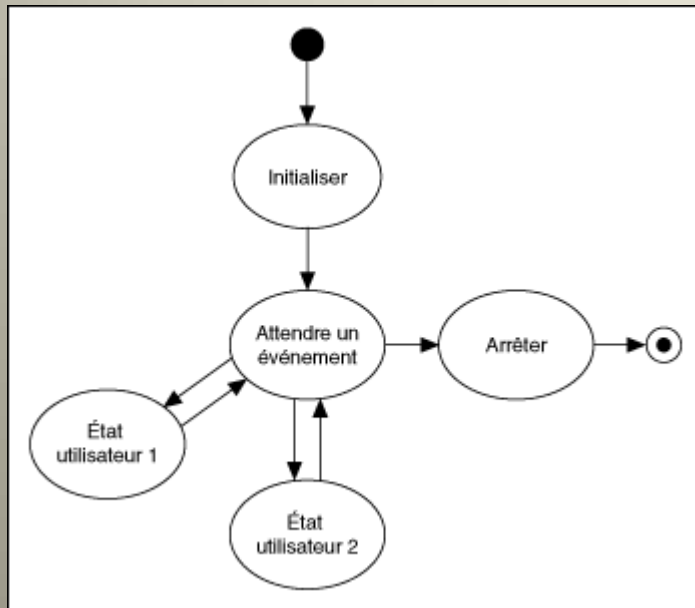
# La machine d'états de JKI

- Rappels sur les machines d'états
- Présentation de la machine d'états de JKI
- Avantages et inconvénients
- Exemple
- Liens

# Rappel sur les machines d'états

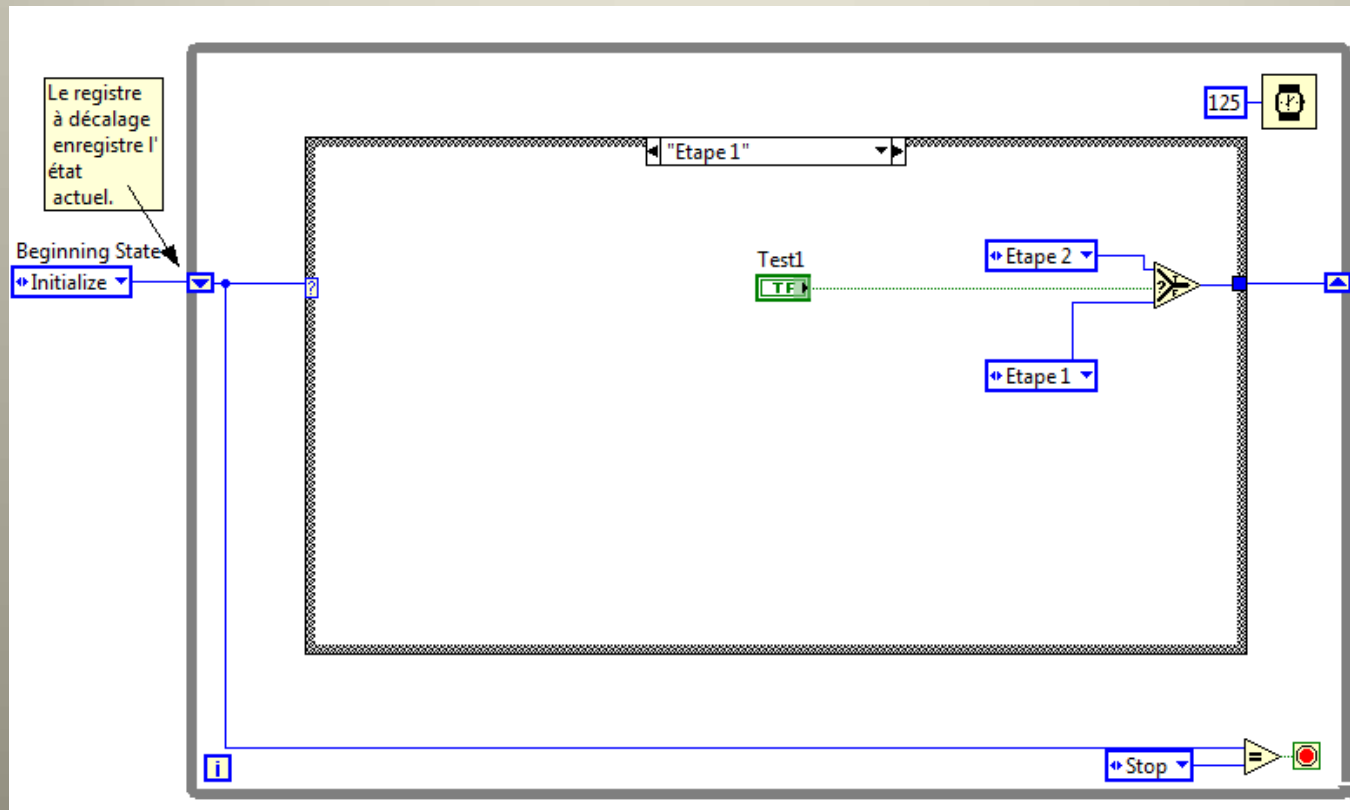
- Définition: Le modèle Machine à états simples facilite la définition de la séquence d'exécution des sections du code. Cette implémentation spécifique, souvent appelée machine de Moore, détermine l'état suivant en fonction de décisions prises dans l'état actuel. **La conception de ce modèle facilite l'insertion de nouvelles sections de code, la suppression de sections de code existantes ou le changement de l'ordre d'exécution des sections, le tout sans devoir modifier considérablement la structure de votre application.**

# Rappel sur les machines d'états



# Rappel sur les machines d'états

- La machine d'états simple de Labview



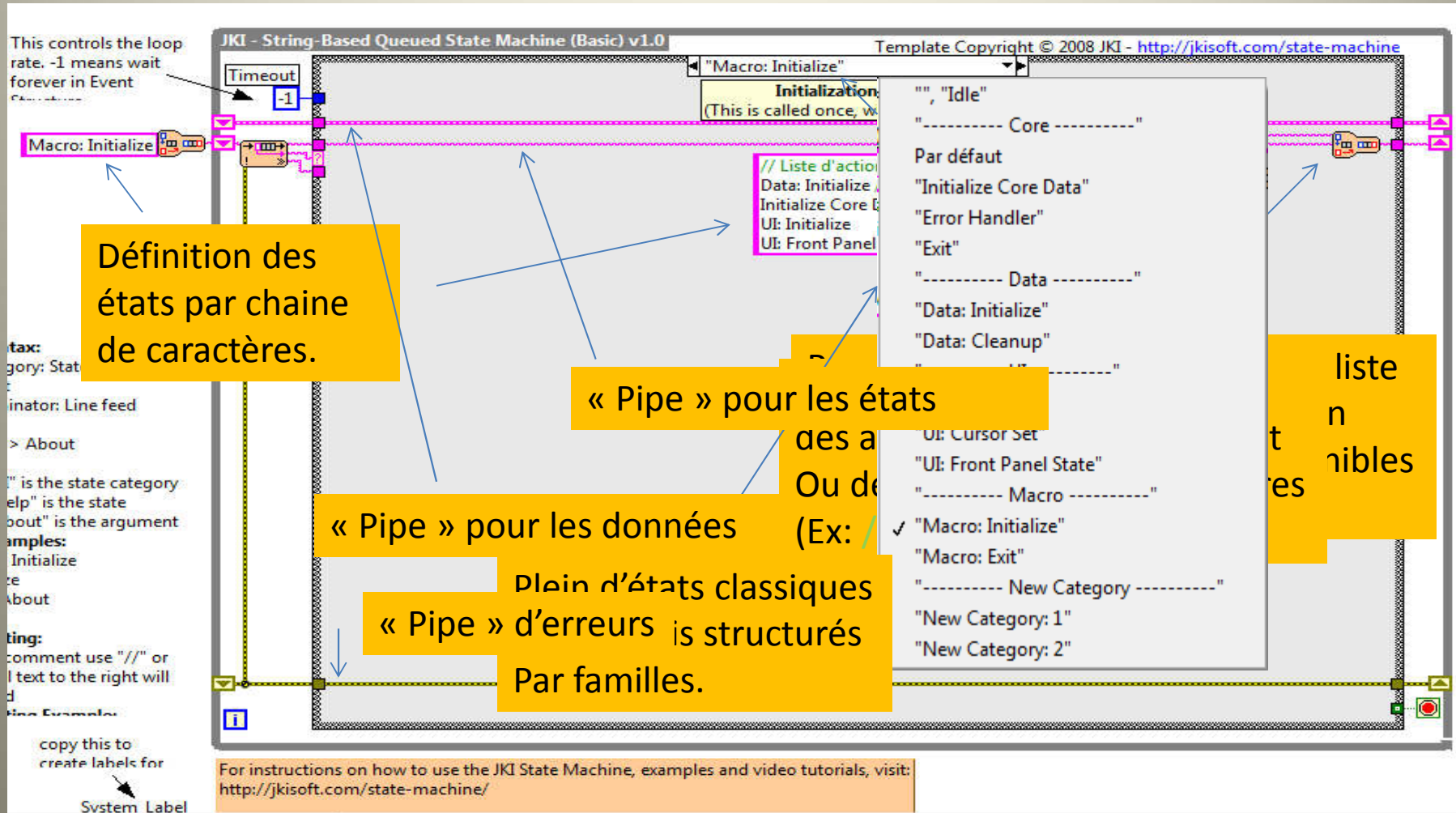
<http://www.ni.com/white-paper/14120/fr>

Eric BOBILLIER INRA  
eric.bobillier@rennes.inra.fr

# Rappel sur les machines d'états

- **Avantage:** Principalement sa simplicité de mise en œuvre et de compréhension.
- **Désavantages:**
  - On ne peut passer que d'un état à un suivant.
  - Il est assez compliqué d'utiliser un état dans plusieurs processus différents
  - Pas de possibilité de passer de paramètres et/ou de commentaires avec la valeur de l'état suivant.
  - Il faut recréer des étapes pour des actions classiquement utilisées

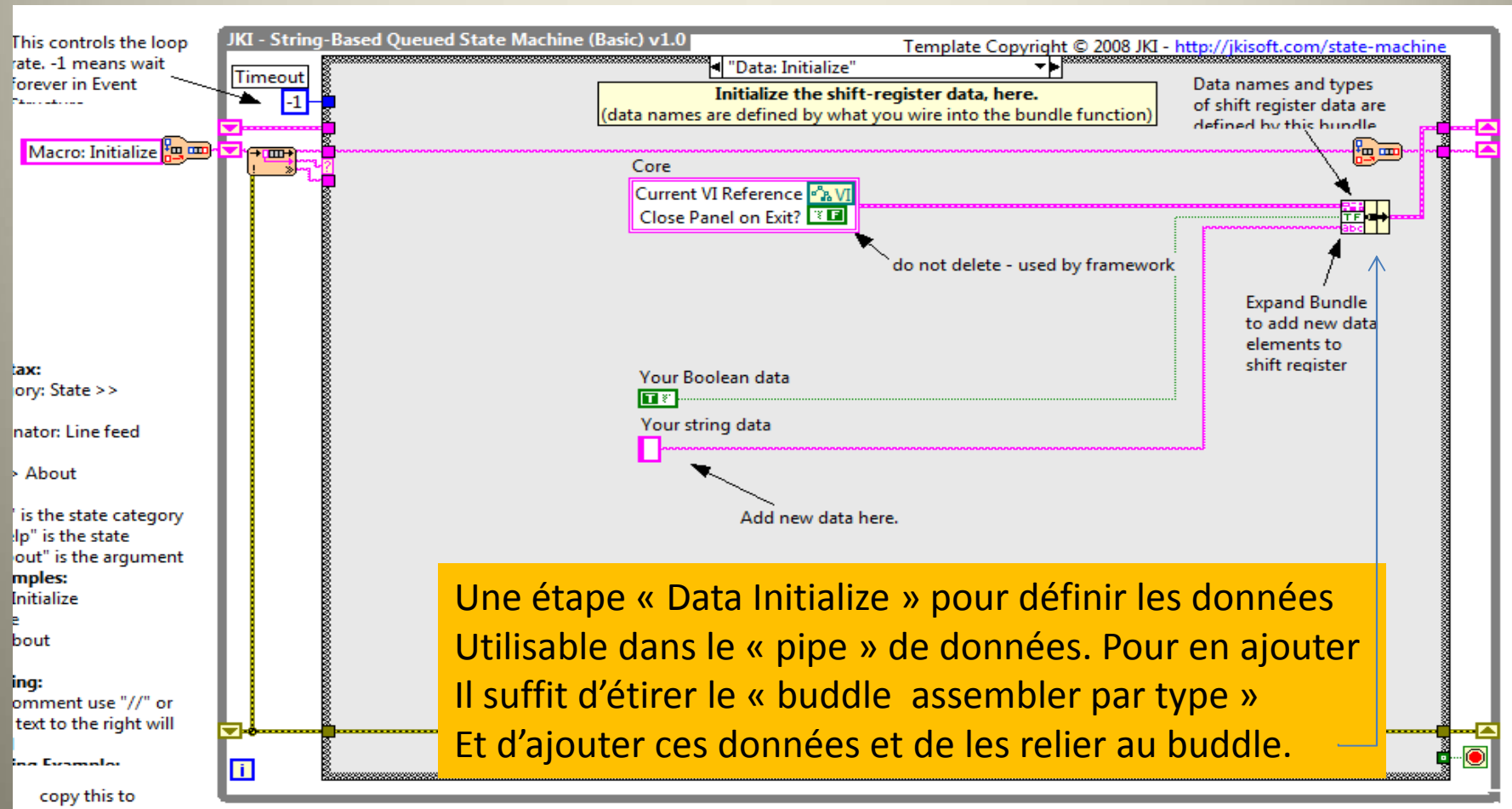
# La machine d'états de JKI



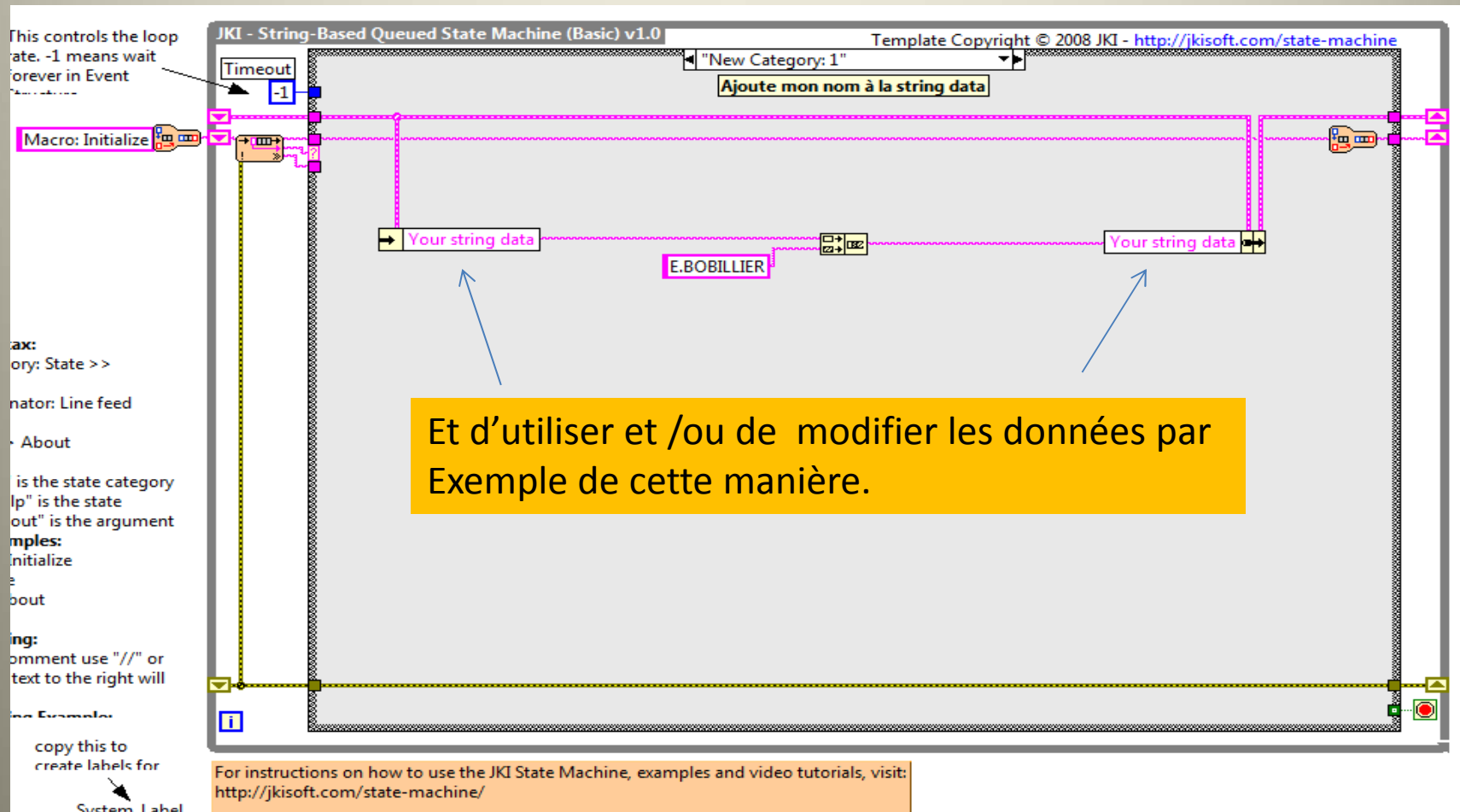




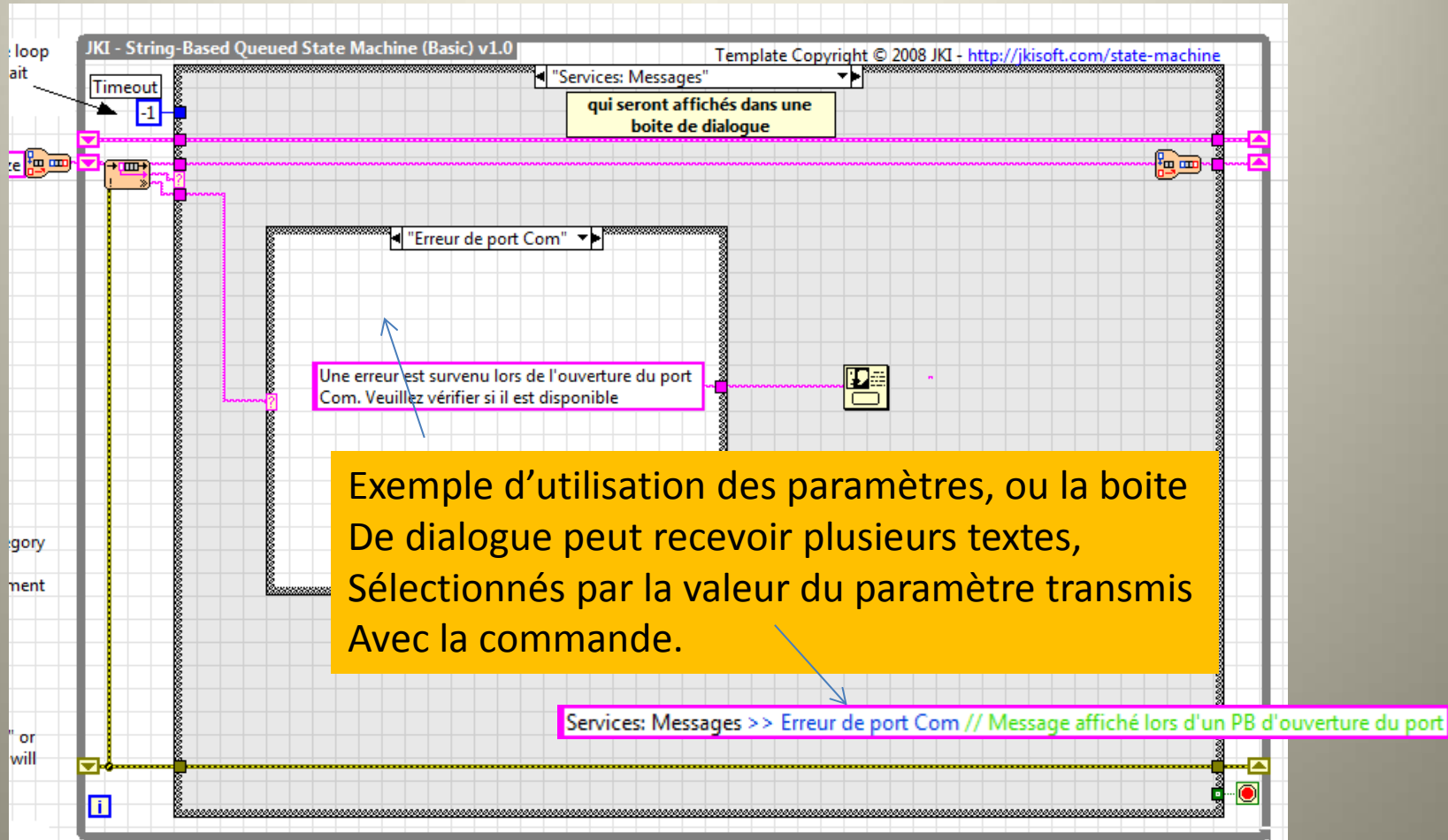
# La machine d'états de JKI



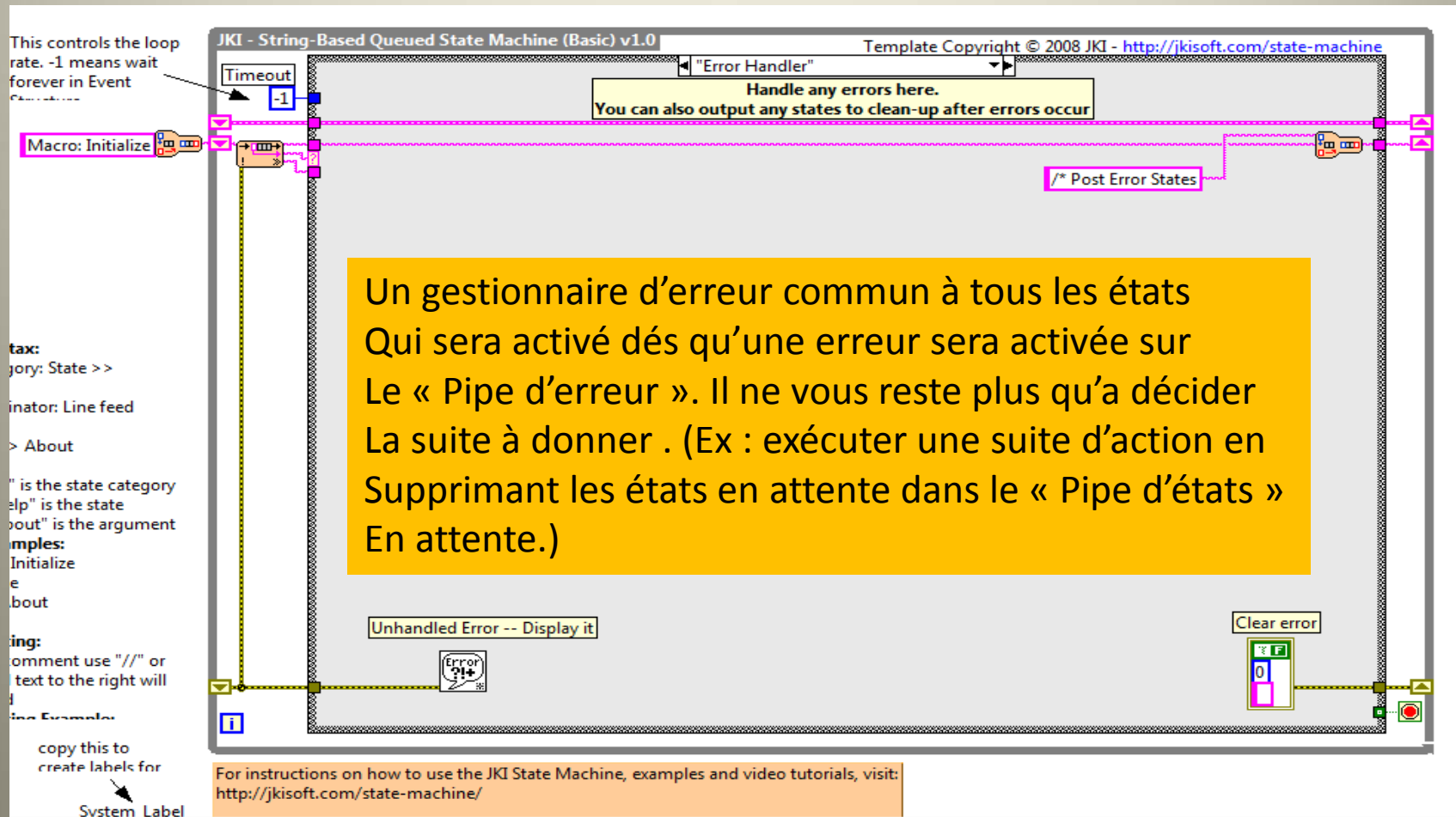
# La machine d'états de JKI



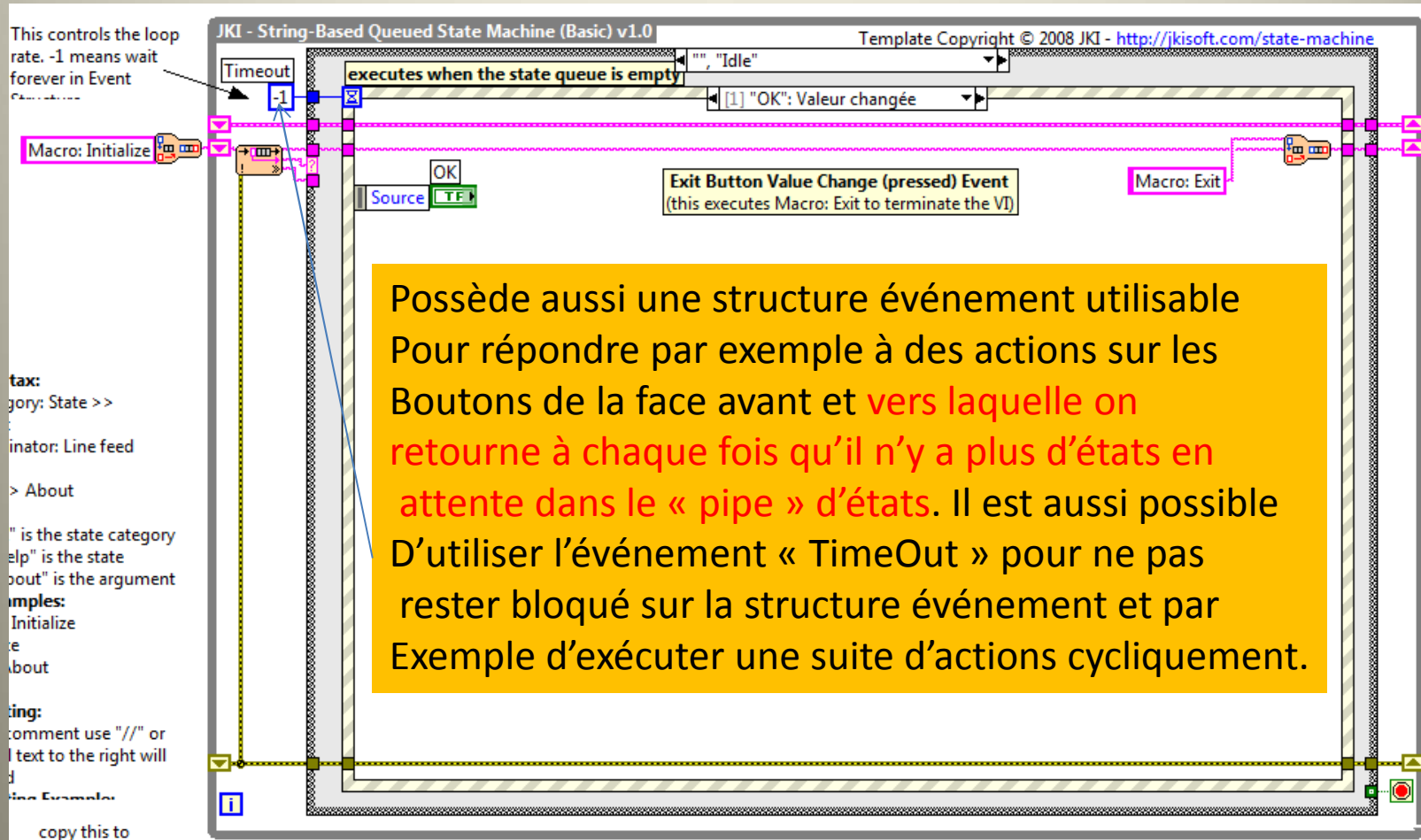
# La machine d'états de JKI



# La machine d'états de JKI



# La machine d'états de JKI



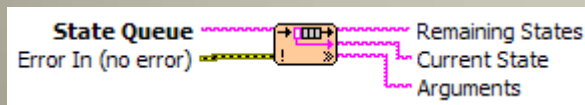
# La machine d'états de JKI

- Cette machine d'état est livrée avec quelques Vi's de support:

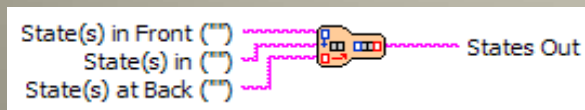
Template de la machine  
D'états.



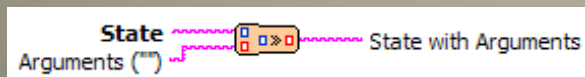
Dossier des vi's support



Ce vi permet de décoder la file des états présent dans Le pipe, et de récupérer le 1<sup>er</sup> état de la file ,ainsi que Ces arguments (paramètres).



Ce vi permet de construire un assemblage d'états.



Ce vi permet d'ajouter un argument à un état.

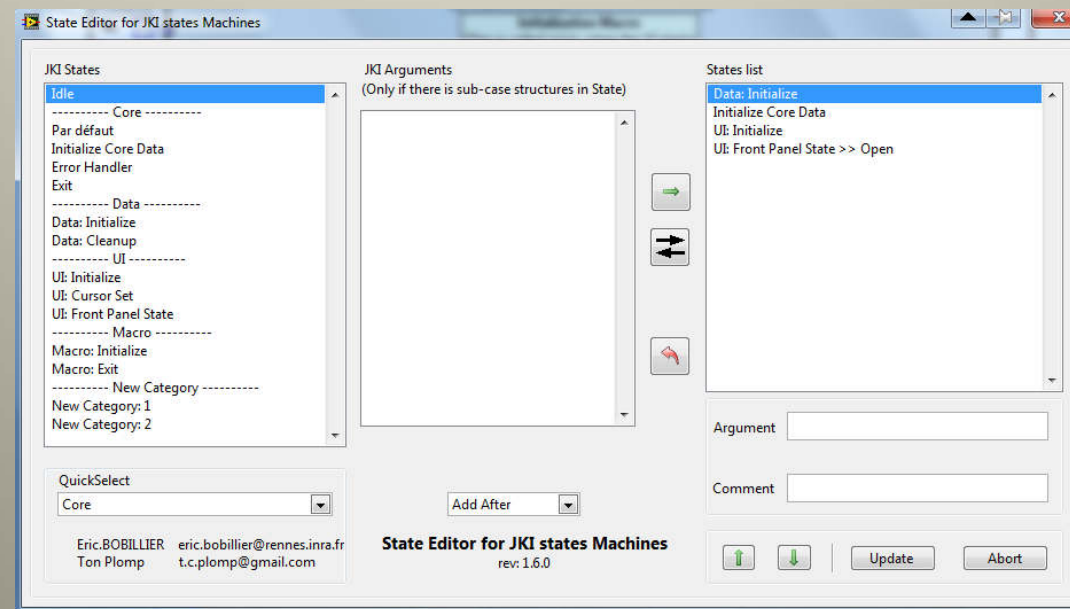
# La machine d'états de JKI

- **Avantages:** Tous ceux développés dans les slides précédentes. De plus il est aussi très facile d'utiliser un état dans plusieurs processus , même si celui-ci à un comportement différent dans ceux-ci.(utilisation des arguments pour le personnaliser. Ex: états « UI: Front Panel State »).
- **Inconvénients:** Comme dans toutes les machines d'états il faut dès qu'elles deviennent complexes, mieux lui adjoindre un graphique ou des explications écrites pour décrire le fonctionnement des processus et des enchainements d'états.

# Outils pour la machine d'états de JKI

- Afin de rendre plus aisée l'utilisation et la mise en œuvre de cette machine d'états j'ai conçu plusieurs outils qui sont regroupés dans un Quick-Drop plugin. Les différents outils sont:
- 1) **State Editor for JKI State Machine** qui permet de créer ou modifier rapidement des déclarations d'états.

(mise en oeuvre: Sélectionnez un constante chaine et faire "Ctrl + Space Barre, puis à l'ouverture de la boite de dialogue :Ctrl + K")





# Outils pour la machine d'états de JKI

- 2) Add\_JKI\_Frame: qui permet de créer rapidement un nouvel état sans avoir à tout recabler systématiquement.
- (mise en oeuvre: Sélectionnez la structure de Cas et faire “Ctrl + Space Barre, puis à l'ouverture de la boîte de dialogue :Ctrl + K”)
- 3) Clean\_JKI\_States\_Machine: Qui permet en fin de développement de supprimer tout les assembleurs d'états inutiles car ne comprenant qu'un état à assembler. Cela permet de diminuer l'empreinte mémoire car ce VI est réentrant.
- (mise en oeuvre: ne rien sélectionner, et faire “Ctrl +Space Barre, puis à l'ouverture de la boîte de dialogue :Ctrl + **Shift** + K”)
- Quick\_Drop que vous pouvez télécharger ici:  
<http://forums.jki.net/topic/2416-qd-for-jki-stm/>  
et qui regroupe ses trois outils en un seul.

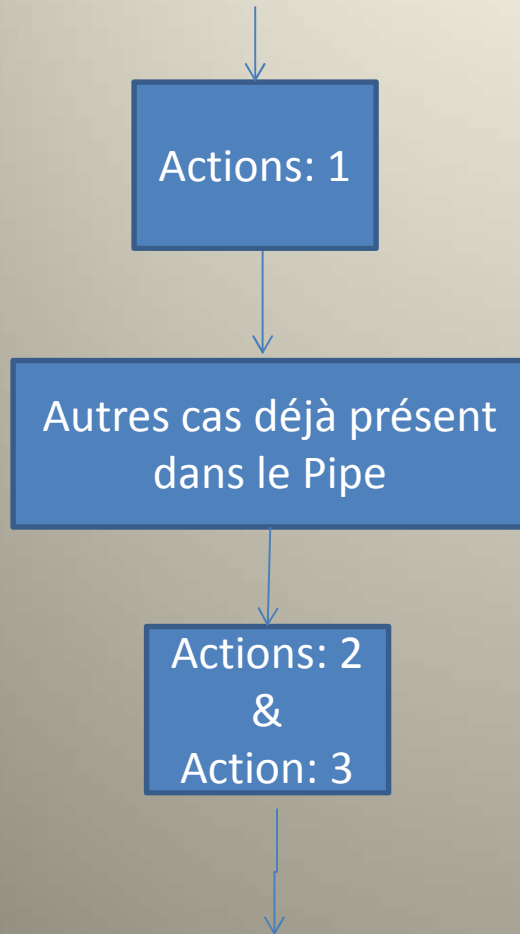


# Liens

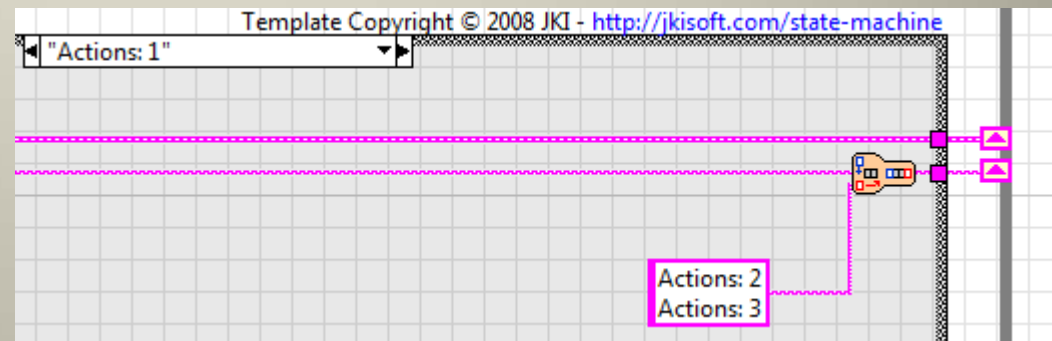


- L'ensemble de ces outils doivent être installés en utilisant un autre outil de JKI « VIPM ».  
<http://jki.net/vipm>
- La machine d'état de JKI et ses tutoriels:  
<http://jki.net/state-machine>
- Quick\_Drop que vous pouvez télécharger ici:  
<http://forums.jki.net/topic/2416-qd-for-jki-stm/>  
et qui regroupe ces trois outils en un seul.

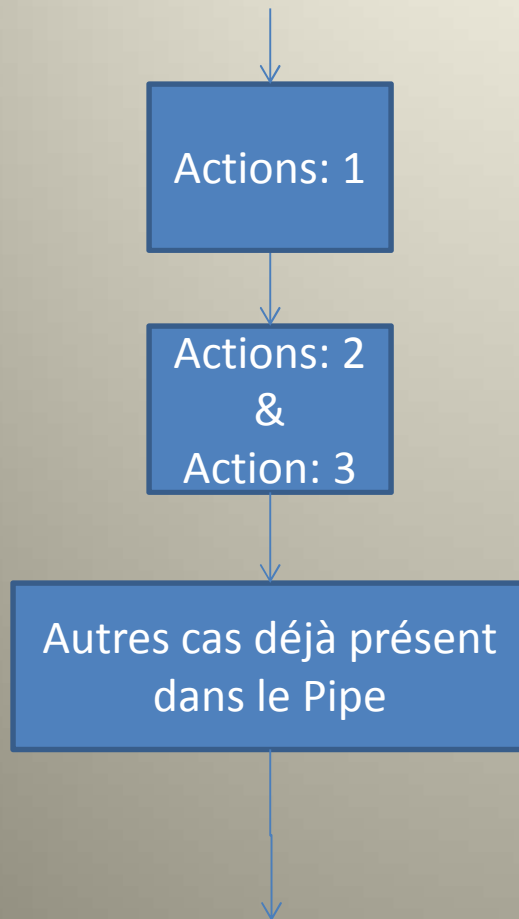
# Exemple d'utilisation classique



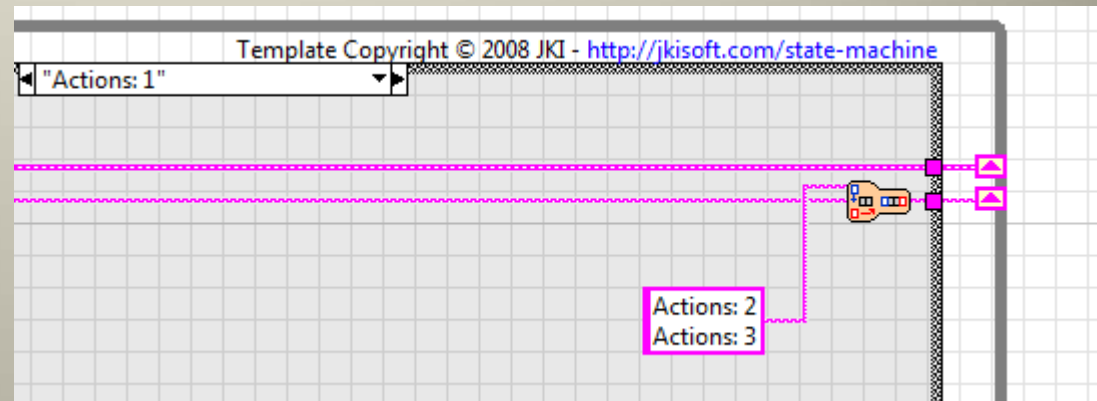
Ajoute après les éventuels cas déjà présent dans le « Pipe d'états » et qui contenait l'action 1, les actions 2 et 3.



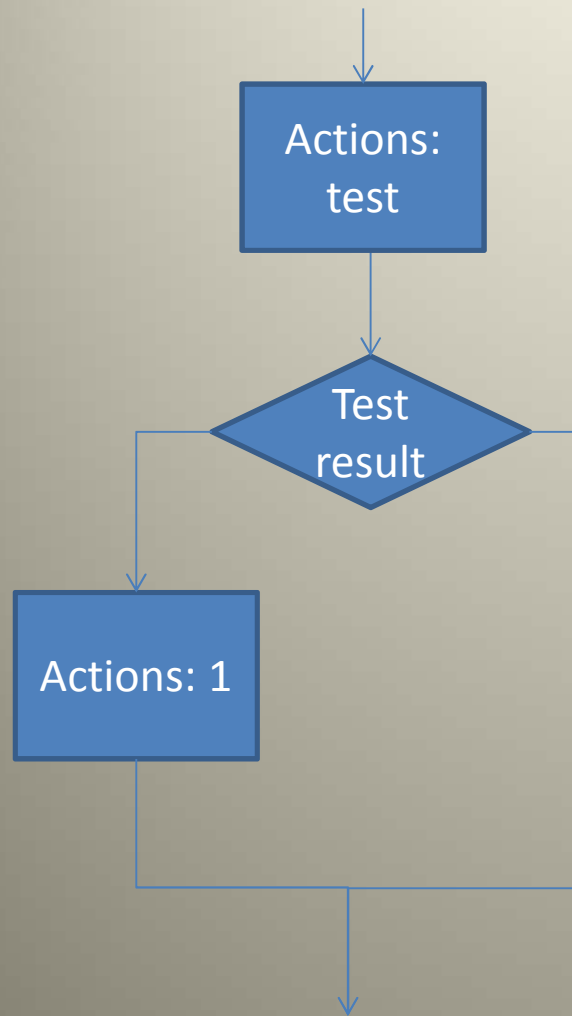
# Exemple d'utilisation classique



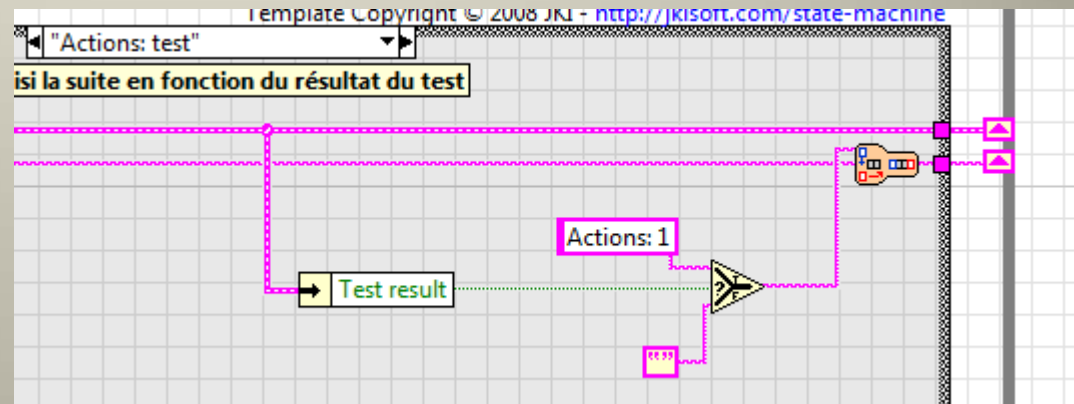
Ajoute **avant** les éventuels cas déjà présent dans le « Pipe d'états » et qui contenait l'action 1, les actions 2 et 3.



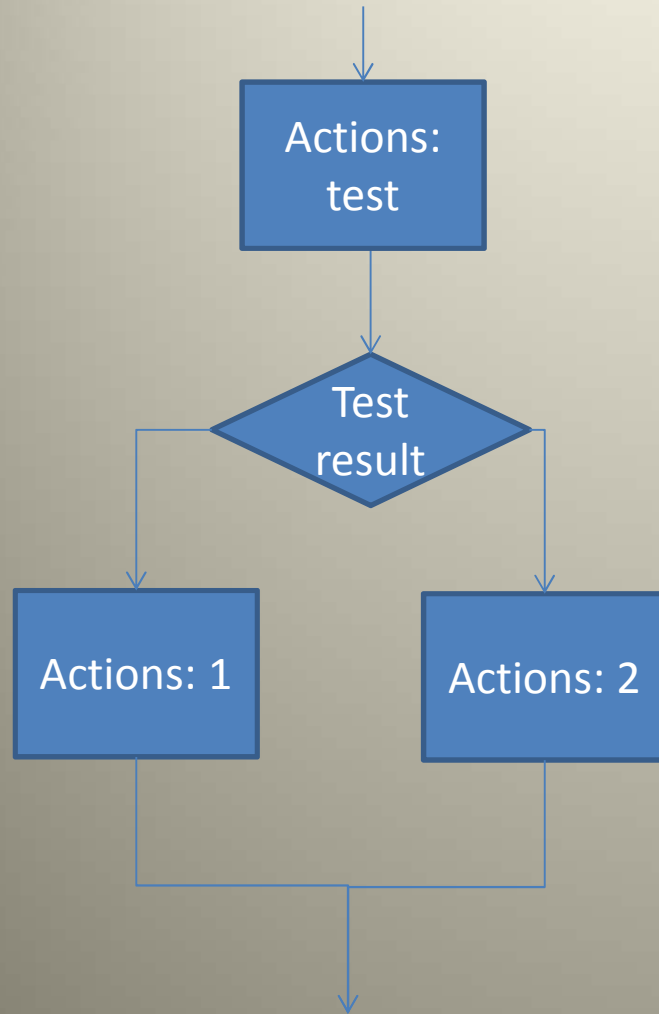
# Exemple d'utilisation classique



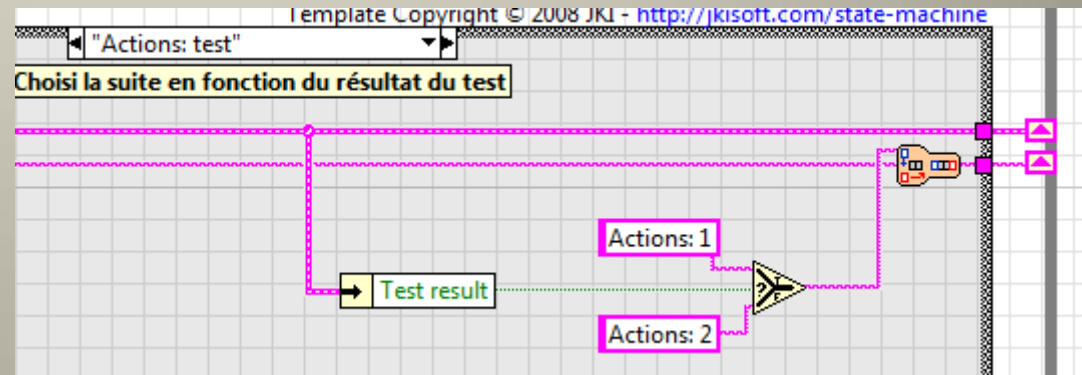
Ajoute aux éventuels cas déjà présent dans le « Pipe d'états», l'action 1 si le résultat du test est positif. Et continu.



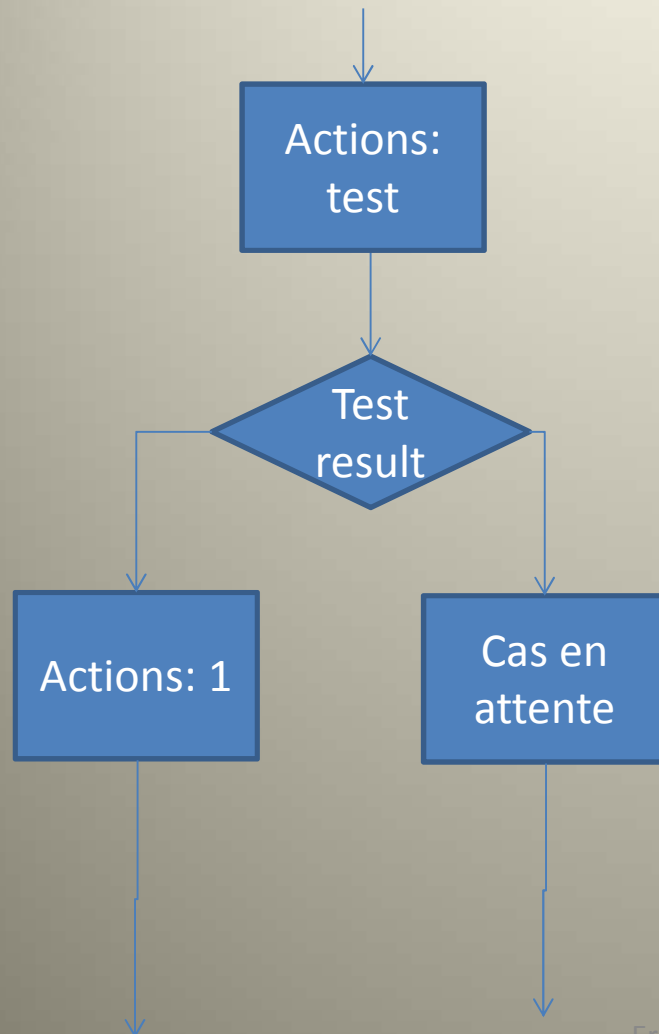
# Exemple d'utilisation classique



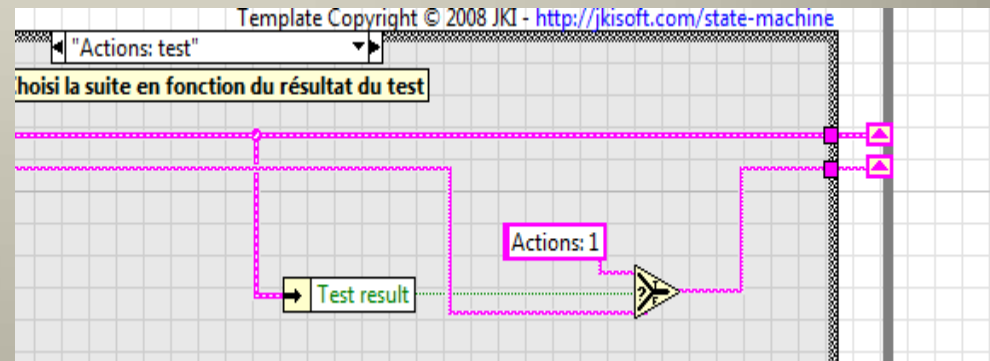
Ajoute aux éventuels cas déjà présent dans le « Pipe d'états» l'action 1 ou 2 selon le résultat du test.



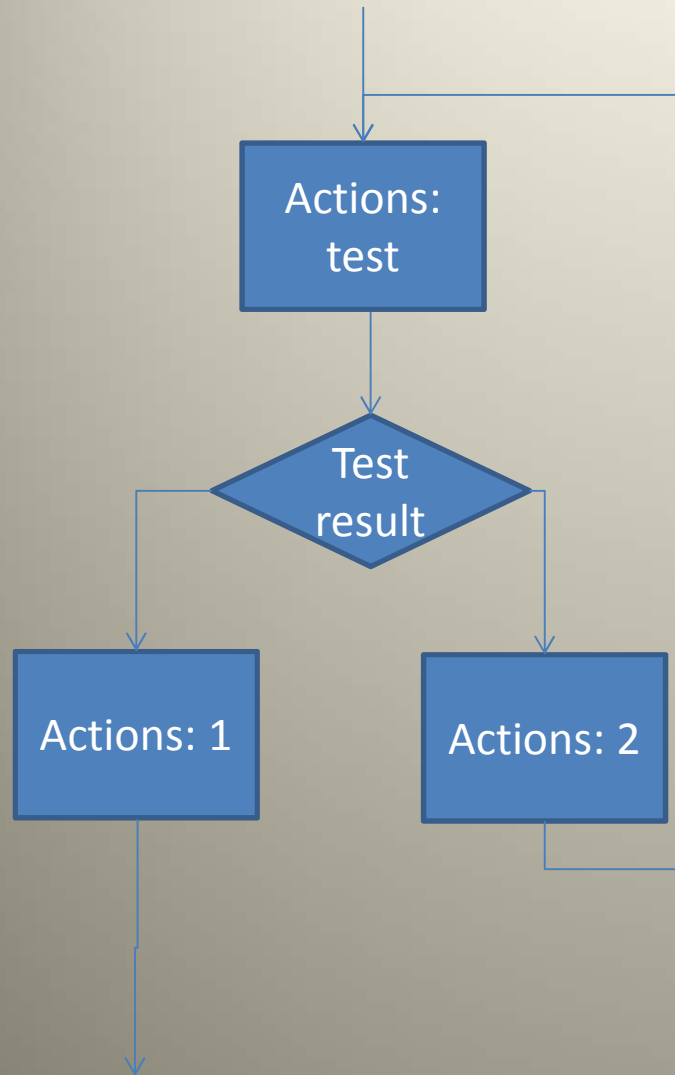
# Exemple d'utilisation classique



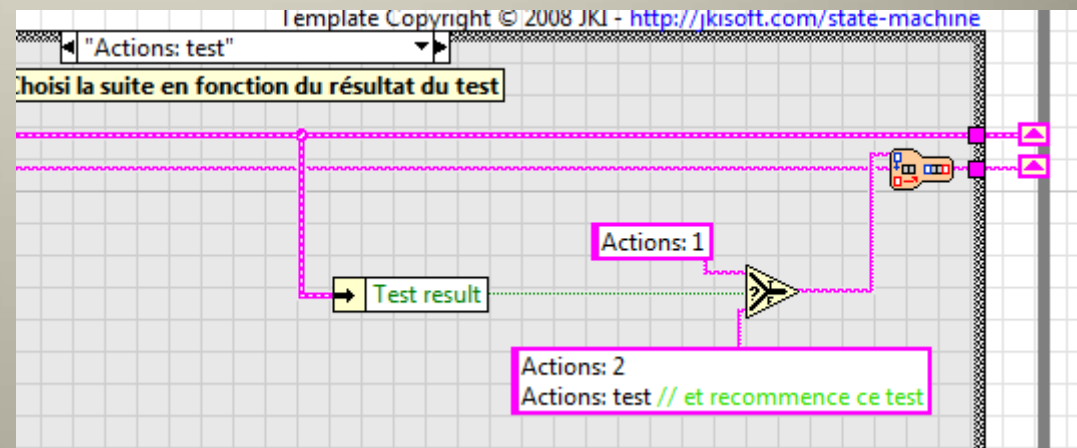
Si résultat du test est négatif , continue les éventuels cas déjà présent dans le « Pipe d'états» . Si positif, alors joue l'action 1 et supprime les cas en attente.



# Exemple d'utilisation classique



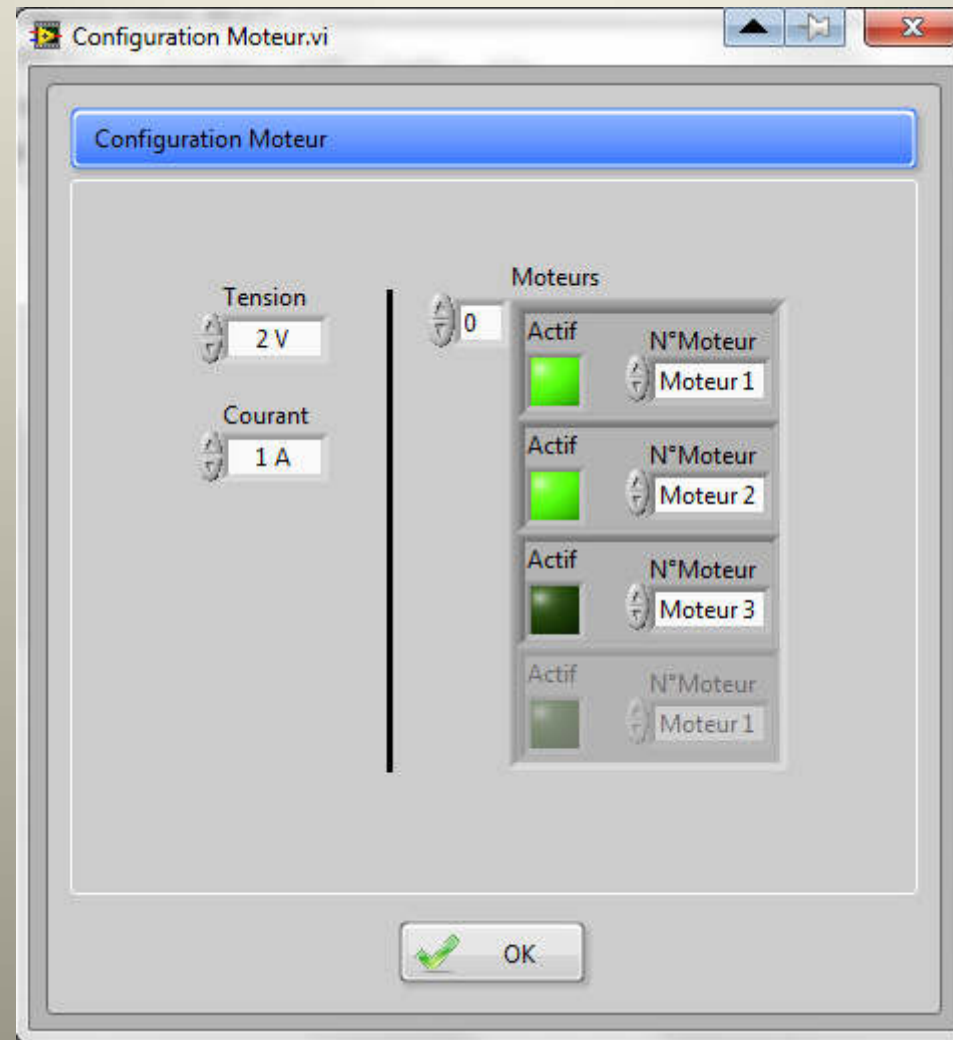
Ajoute aux éventuels cas déjà présent dans le « Pipe d'états» l'action 1 si test positif et si négatif ajoute l'action 2 et recommence ce test.





# DEMO

# Exemple: L'interface de configuration



# Exemple: L'interface de configuration

- Dans un programme , il est très fréquent que l'on ai besoin d'une ou plusieurs interfaces de configurations pour définir un certain nombre de paramètres qui seront utilisés dans son déroulement. Ces paramètres, doivent bien entendu être accessible à l'utilisateur pour pouvoir être modifiés et sauvegardés (ceci est le cas le plus fréquent).

J'appellerai ce cas « Edit »

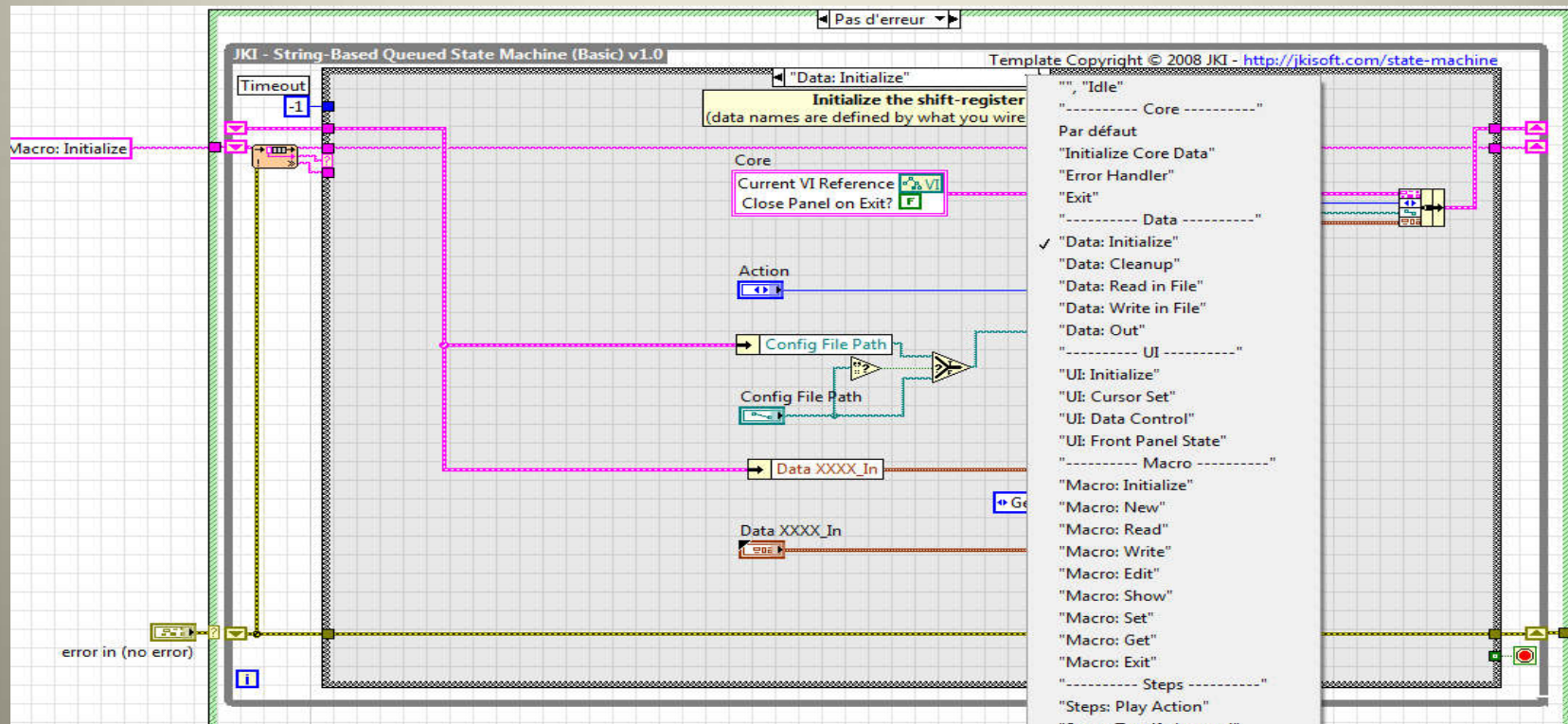
# Exemple: L'interface de configuration

- Mais il existe plusieurs autres cas ou ces paramètres peuvent manipuler pour répondre à des besoins particuliers:
- 1) Le cas L'on a besoin de (re)-partir sur des valeurs de base (valeur par défaut). Cas « New »
- 2) Le cas ou l'on a besoin de charger les valeurs précédemment enregistrées, si elles existent, mais sans ouvrir la face avant du vi de configuration. Cas « Read »
- 3) de même on peut avoir besoin d'enregistrer les paramètres mais sans ouvrir la face avant du vi de configuration. Cas « Write »
- 4) il se peut aussi, que l'on souhaite afficher les paramètres, mais sans autoriser leurs modifications. Cas « Show »

# Exemple: L'interface de configuration

- 5) De même on peut vouloir modifier temporairement les paramètres, mais sans enregistrer ces modifications. Et ce à la manière d'une globale fonctionnelle qui pourrait être modifiés à un endroit du programme et relut dans un autre endroit. Cas « Get » et « Set ».
- Sachant que tous ces cas d'utilisation utilisent beaucoup de parties communes, j'ai décidé de concevoir un template autour d'une machine d'état de JKI et fonctionnant aussi à la manière d'une globale fonctionnelle. L'intérêt d'utiliser un template est que la structure restera toujours la même, seule les données (paramètres) traités changeront...

# Exemple: L'interface de configuration



Démo d'utilisation de ce template. (Demo utilisation template configuration.vi)

# Présentation de l'outil Config creator.

- Comme indiqué précédemment, dans un fichier de configuration créé autour de mon template, seules les données changent, Il faut donc simplement changer les données d'entrée et de sortie (connecteur) et celle affichées sur la face avant. Afin de rendre plus aisées leurs utilisations dans le template, les paramètres doivent être regroupés dans un cluster (def type strict). Il faut ensuite remplacer les contrôles d'entrée et de sortie et celui du contrôle affiché, par ceux nouvellement créés. Puis remettre à l'échelle, les différentes décorations du template, afin de d'obtenir un fichier de configuration final, prêts à l'emploi.
- Afin de limiter le côté fastidieux de cette tâche, j'ai conçu l'outil Config creator qui permet de le faire en quelques secondes. Démo....